

SIMI-003/01US



COPY OF PAPERS
ORIGINALLY FILED

#5

PATENT

METHOD AND SYSTEM FOR A GRAPHICAL

REAL TIME FLOW TASK SCHEDULER

The present application claims priority from U.S. Provisional Patent Application No. 60/105,764 filed on October 26, 1998. The contents of that application, in its entirety, is
5 hereby incorporated by reference.

Background of the Invention

Field of the Invention

The present invention is related to systems and methods for providing a real time task scheduler for material processing systems. Specifically, the present invention relates to methods and systems for a graphical task scheduler for a manufacturing plant.

Description of the Related Art

Process industries, including but not limited to refining, chemical, petrochemical, and pharmaceutical processing, have experienced a need to operate ever more efficiently in order to remain competitive. This need has resulted in the development and deployment of process 15 modeling systems. A process engineer uses these modeling systems to build a process model, or flowsheet, of an entire processing plant using equipment models provided by the modeling system. These flowsheet models are used to design and evaluate new processes, redesign and retrofit existing process plants, and optimize the operation of existing process plants. Typically, the process engineer must first build the flowsheet model, compile the model, and perform model 20 checking before any model errors are detected.

Conventional modeling systems typically provide simulation, data reconciliation, and optimization of plant processes using corresponding separate programs and separate models.

Ongoing computer and software technology advances have enhanced the modeling systems' capabilities. The process engine is now able to model larger and more complex processes with greater detail and precision.

Conventional modeling systems are complex and require a user to program the plant
5 model into a computer. However, typically the process engineers associated with the design and optimization of a plant do not possess computer programming skills, while computer programmers do not possess the knowledge of the plant. The amount of time and energy lost in interfacing the process engineer with the computer programmers is substantial.

Once a plant is fully designed and optimized, it is often desirable to automate many of the features of the plant. To automate the plant, the process engineer would again interface with the computer programmer to develop a second software package. What is needed is a system where the modeling, optimization, and automation processes can be performed from a single software package. Further, applications should be simple and intuitive to construct without an in-depth knowledge of computer programming.

15 Summary of the Invention

The present invention is a real time, on-line, optimization tool for refineries and plants, typically petroleum refineries and petrochemical plants. The present invention provides a unified modeling environment for both on-line and off-line applications, as well as a rigorous data reconciliation capability.

20 The present invention includes a graphical user interface allowing a user to construct real time sequences through a point-and-click interface. The user selects a task from a list and inserts

the task into the appropriate location in the sequence. The user continues to add tasks to the sequence until the sequence is complete.

Each real time sequence is associated with a list of tasks whose execution can be scheduled or invoked on demand. In one embodiment, the tasks in the sequence are executed 5 synchronously in a predefined order. For scheduled sequences, the user defines the schedule of execution. In another embodiment, there are two kinds of schedules that the user may choose from: schedules whose frequency is less than a day and those whose frequency is more than or equal to a day.

For schedules having a frequency of less than a day, the user enters a start time and a delta time. The delta time is the amount of time between scheduled execution and can be any amount of time less than 24 hours. The start time gives the basis from which the schedule is generated.

In one embodiment, for schedules having a frequency of more than a day, the user can choose a daily, weekly, or monthly based schedule. In any case, the user generally enters a start 15 time, which specifies the time of day each execution is to begin, and effective date. For a daily schedule, the user also specifies the number of days between executions. For a weekly schedule, the user specifies the day of the week and the number of weeks between repetitions of the schedule. For a monthly schedule, the user specifies the day of the month and the number of months between executions.

20 Real time sequences are registered with a scheduler to be executed on a designated schedule. On-line sequences which are scheduled are initiated by the scheduler according to the defined on-line sequence schedule. On-line sequences which are unscheduled will not be

scheduled but may be executed on demand. For example, a task in a scheduled sequence may demand the execution of an unscheduled sequence.

In one embodiment, on-line sequences have a limit time, which is the maximum amount of time that the sequence can take to execute. On-line sequences which continue to execute past the limit time are halted or killed by the scheduler.

A sequence may be designated as a model sequence for performing on-line optimization. These sequences are used to model the plant and perform optimization analysis. Once a model is optimized, the sequences may be used to control the plant operation.

The tasks of each sequence may include branching logic. This allows any task to have more than one exit branch. The user can define which exit branch is to be used based on a variety of factors.

Brief Description of the Drawings

These and other features and advantages of the invention will become more apparent upon reading the following detailed description and upon reference to the accompanying drawings.

Figure 1 illustrates the components used in a real time system according to one embodiment of the present invention;

Figure 2 is a diagram showing the components of the real time scheduling software according to one embodiment of the present invention;

Figure 3 is a sample screen view of a sequence development window as used in the real time scheduling software according to one embodiment of the present invention;

Figures 4A-4D illustrate the manner in which a task may be added to a sequence according to one embodiment of the present invention;

Figure 5 is a sample screen view of a dialog box used to establish parameters for a sequence according to one embodiment of the present invention;

5 Figure 6A is a sample screen view of a dialog box used to schedule a sequence when the sequence occurs more than daily;

Figure 6B is a sample screen view of a dialog box used to schedule a sequence when the sequence occurs daily;

Figure 6C is a sample screen view of a dialog box used to schedule a sequence when the sequence occurs weekly;

Figure 6D is a sample screen view of a dialog box used to schedule a sequence when the sequence occurs monthly;

Figure 7 is a sample screen view of a dialog box used to establish parameters for a task according to one embodiment of the present invention;

15 Figure 8 is a sample screen view of a dialog box used to program a script to create a custom task according to one embodiment of the present invention;

Figure 9 is a sample screen view of a dialog box used to establish parameters for a task to run an external program according to one embodiment of the present invention;

20 Figure 10 is a sample screen view of a dialog box used to establish parameters for determining steady state detection according to one embodiment the present invention; and

Figure 11 is a sample sequence that may be created according to one embodiment of the present invention.

Detailed Description of a Preferred Embodiment

The network architecture for one embodiment of the present invention is indicated generally at 10 in Figure 1. (In Figure 1 and throughout the drawings, components which correspond to components shown in previous figures are identified using the same reference numbers.) The system 10 may include a local area network (LAN) 102 which is connectable to other networks 104, including other LANs or portions of the Internet or an intranet, through a router 106 or similar mechanism. One example of such a LAN 102 may be a process control network 105 to which process control devices including process control computers are connected. Process control networks 105 are well known in the art and are used to automate industrial tasks. The network 104 may be a corporate computing network, including possible access to the Internet, to which other computers and computing devices physically removed from the plant floor are connected. In one embodiment, the LANs 102, 104 conform to Transmission Control Protocol/Internet Protocol (TCP/IP) and Common Object Request Broker Architecture (COBRA) industry standards. In alternative embodiments, the LANs 102, 104 may conform to other network standards, including, but not limited to, the International Standards Organization's Open Systems Interconnection, IBM's SNA®, Novell's Netware®, and Banyan VINES®.

The system 10 includes a server 108 that is connected by network signal lines to one or more clients 112. In one preferred embodiment, the server 108 includes a DEC Alpha® NT operating system. In an alternative embodiment, the server 108 includes a UNIX operating system. In one preferred embodiment, the client 112 includes a Windows® NT operating system. The server 108 and clients 112 may be configured by those of skill in the art in a wide variety of ways to operate according to the present invention. The server 108 and clients 112

may be uniprocessor or multiprocessor machines. The server 108 and clients 112 each include an addressable storage medium such as random access memory and may further include a non-volatile storage medium such as a magnetic or an optical disk.

The system also includes a storage medium 110 that is connected to the process control network 102 or corporate control network 104. In one embodiment, the storage medium 110 may be configured as a database from which data can be both stored and retrieved. The storage medium 110 is accessible by devices, such as servers, clients, process controllers, and the like, connected to the process control network 102 or the corporate control network 104.

2001 SEPTEMBER 10

Suitable servers 108 and clients 112 include, without limitation, personal computers, laptops, and workstations. The signal lines may include twisted pair, coaxial, telephone lines, optical fiber cables, modulated AC power lines, satellites, and other data transmission media known to those of skill in the art. A given computer may function both as a server 108 and as a client 112. Alternatively, the server 108 may be connected to the other network 104 different from the LAN 102. Although particular computer systems and network components are shown, 15 those of skill in the art will appreciate that the present invention also works with a variety of other networks and components.

Figure 2 illustrates the process overview 200 of the real time subsystem according to one embodiment of the invention. A user interfaces with the real time set subsystem through a GUI client 205. The GUI client 205 may be one of the clients 112 from Figure 1. The graphical user 20 interface provided by the GUI client 205 provides a user-friendly method for a user to interact with the real time subsystem. The specifics of the GUI client 205 will be discussed below.

In one embodiment, sequences constructed by the GUI client 205 are processed to a server 210 and saved in corresponding sequence databases 245. Thus, by way of example, each separate sequence created by the GUI client 205 has a separate sequence database 245. When a scheduler 225 signals a sequence execution process 220 to run a specific sequence, the sequence 5 execution process 220 retrieves the sequence from the corresponding sequence database 245 to determine the steps to run. In one embodiment, the GUI client 205 exchanges information with the server 210, including for example, the data input by a user. The scheduler 225 has read access to the sequence database 215. When requested by the GUI client 205, the server 210 directs the scheduler 225 to put sequences on-line or take sequences off-line. When a sequence 10 is on-line, it is ready to run under the direction of the scheduler 225. When the server 210 requests the scheduler 225 to place the sequence on-line, the scheduler 225 opens a sequence execution process 220 to control that particular sequence. Therefore, multiple sequence execution 220 processes may be open at any given time.

The scheduler 225 also communicates with one or more timers 230. The scheduler 225 15 may request an alarm to be set in the timer 230. The timer 230 communicates back to the scheduler 225 when the alarm is tripped. These alarms may include, for example, a calculated end of a sequence time or a calculated start time of a new sequence. The timer 230 can monitor multiple alarms simultaneously.

The scheduler 225 maintains three collections of pointers to on-line sequences: on-line, 20 scheduled, and running. The on-line collection may be sorted by name and contains pointers to the associated sequence execution process 220. The scheduled collection is sorted by next runtime and contains those sequences that are on-line and scheduled. The running collection is

sorted by next kill time and contains sequences that are on-line and running. Periodically, the scheduler 225 checks the scheduled and running collection for the next event. When the time occurs for a sequence to be started, the scheduler signals the appropriate sequence execution process 220 to run the sequence. The schedule 225 monitors the sequence status in the sequence database 215 to determine when the sequence has completed execution.

When a sequence execution process 220 receives a start signal from the scheduler 225, the sequence execution process 220 begins executing the sequence and sets the status of the sequence in the sequence database 215 to running. Upon completion of the sequence, the sequence execution process 220 sets the status of the sequence back to waiting in the sequence database 215. If the execution time of a sequence exceeds the limit and needs to be killed, the scheduler 225 transmits a kill signal to the sequence execution process 220. The sequence execution process 220 then stops and resets the sequence. The sequence execution process 220 then sets the status of the sequence back to waiting in the sequence database 215.

In one embodiment, during execution of a sequence, the sequence execution process 220 may write messages to the message server 235. The message server 235 then writes the messages to a message log database 240 for access by the server 210. The sequence execution process 220 continues running the sequence while the messages are processed. To allow multiple sequence execution processes 220 to run at any given time, the message server 235 handles all messages from the sequence execution processes 220 on a first-come, first-serve basis.

Advantageously, when the GUI client 205 creates a sequence designated as a model sequence, the server 210 communicates with a model application server 250. The model

application server 250 stores the model sequences in a model application database 255. The model application sequences are used to simulate real time sequences, and can be used to optimize the process. Details on optimization and modeling of material process systems are included in the co-filed applications entitled INTERACTIVE PROCESS MODELING SYSTEM
5 and PROCESS MODEL GENERATION INDEPENDENT OF APPLICATION MODE filed on even date herewith and which are hereby incorporated by reference in their entirety.

An example of the GUI interface 205 according to one embodiment of the present invention is shown in Figure 3. For each sequence created, a sequence process flow diagram (PFD) window 300 is opened. In this embodiment of the PFD window, the name of the sequence is displayed in the identification bar 305. In one embodiment, each sequence has a unique identifying name. A series of pulldown menus 310 and a button bar 315 are provided for the user to interface with the PFD window 300. The use of pulldown menus 310 and button bars 315 are well known in the art and therefore do not require further explanation herein. The PFD window 300 also includes a sequence display window 340 which provides a graphical display of
15 the current sequence to the user. When the sequence PFD window 300 is first opened, the sequence display window 340 is blank. The user creates a sequence by using a keyboard, a mouse, or other pointing device such as a trackball or joystick to drop and drag tasks from the task palette 320 into the sequence display window 340. The tasks are selected from one of the several task palettes 320. To add the task to the current sequence, the user drags the task into the
20 proper location in the sequence display window 340. The task palette 320 is subdivided into several categories. For example, in a general task window 325, basic tasks such as an input task, an output task and a custom task are displayed. Details of specific types of tasks are discussed

below. In a model task window 335, tasks appear that would be used for a model application sequence. These are tasks such as load case, store case, and solve, which would not be used to control a plant, but would be used to simulate the control of a plant. The generic task window 330 displays tasks used during generic sequences. In one embodiment, the sequence display window 340 provides a continual visual display to the user of the tasks included in the current sequence.

An example of a process of creating and modifying a sequence using the sequence PFD window 300 is shown in Figures 4A-4D. In Figure 4A, an initial sequence is created which includes a start state 405 followed by a Task A 410 and a Task B 415. An exit branch 420 of Task B is a terminal exit branch which causes the sequence to stop running. In Figure 4B, the user has selected and dragged a third Task C 425 having two terminal exit branches 430 and 435 into the sequence display window 340. Task C 425 is selected from one of the task palettes 320. In Figure 4C, the initial sequence is connected to the new Task C 425 by a line 440. The line 440 is created by dragging the terminal exit branch 420 from Task B 415 onto Task C 425. By connecting Task C 425 to Task B 415, the terminal exit branch 420 of Task B 415 is deleted. At this point, the sequence has two terminal exit branches, 430 and 435, from Task C 425. In the present example, Task C 425 contains branching logic 15 which, under certain conditions, would revert the user back to Task B 415. To establish this branch, the terminal point 435 from Task C 425 is connected back to the input of Task B 415 as shown in Figure 4D. This creates a recycle loop 445 in the sequence. Now the sequence in Figure 4D contains the third Task C 425 with one terminal exit branch 430. The details of each task can be displayed and modified by

double-clicking on the respective task icon in the sequence window 340 as discussed in detail below.

Thus, the sequence created in Figures 4A-4D includes the recycle loop 445. By having more than one output branch in a task, a sequence is able to advantageously employ branching logic. Branching logic may be used, for example, to continue to refine modeling estimations in a task until all the data is reconciled. Without branching logic, the task could end without any guarantee the data is reconciled. Another example of the usefulness of branching logic is when interactions are used to optimize the sequence. Without branching logic, the task needed to estimate the number of iterations necessary before the sequence is optimized. By including branching logic, the task can be repeated until a specified condition, in this case the sequence being optimized, is obtained.

The sequence PFD window 300 also assists the user in creating sequences by providing feedback to the user on the status of the sequence. In one embodiment, each task is visually coded to indicate to the user the state of the task. In one example, the coding is a color code. For example, if a task is not fully specified or has no entry point, the color of the task border may be red. Thus, in the examples in Figures 4A-4D, Task C in Figure 4B would have no entry point and therefore would be bordered in red. If a task is fully specified and ready to run, the task border would be black. When each task has a black border, the user knows the sequence is fully specified and ready to run. A task which has been determined to be inactive may be bordered in gray. Other types of coding may include text messages indicating the task state.

The visual coding is also useful during execution of the sequence to indicate to the user the status of the sequence execution. In one example, upon initialization of the sequence, all the

task borders are set to white. When a task in a sequence is currently running, its border is set to green. After successful execution of a task, the border may be changed to blue. If a task was terminated abnormally, the border may be set to red indicating error for the user to investigate.

Once all the tasks in a sequence have been fully specified, the sequence can be initialized.

- 5 Clicking on the setup button on the toolbar 315 of Figure 3 causes a dialog box 500 similar to that as shown in Figure 5 to be displayed on the screen. A title bar 501 of the dialog box 500 contains the name of the sequence. Below the title bar 501 are a series of tabs used to modify the sequence setup. The dialog box 500 shown when a general tab 502 is selected is shown in Figure 5. Selecting the general tab 502 displays a scheduling box 505. In the scheduling box 505, the user has the option of selecting whether a sequence is to be scheduled or not scheduled. If a sequence is scheduled, then the scheduling information for the sequence is used to determine when the sequence is executed. The scheduling information will be described below in further detail. If the sequence is not scheduled, then the sequence may be demand-activated by a task within another sequence. For example, a sequence for data reconciliation may not be scheduled
- 15 but may be activated by another sequence or task which detects the process to be at steady state. A run information box 510 displays the current run number of the sequence. The run number is incremented automatically each time the sequence is executed. This number is used for creating unique objects and output file names for each run of the sequence. For example, data files on any sequence run may be saved using a filename which includes the run number. This ensures
- 20 each data file has a separate name.

All on-line sequences may write messages to an on-line message log. A message logging box 515 allows the user to select the level of messages for the sequence to write to the on-line

message log. For example, the message level may be set to, among other things, brief, detailed, no message logging, or logging exceptions. The selected level can be overridden by individual tasks within the sequence, but serves as a default for all tasks.

If the sequence is to create an output log, an output log box 520 allows the user to specify

- 5 the number of cycles to keep in an output log before the log is restarted. This allows the user to control the amount of memory taken by the output log.

In one embodiment, when a sequence is scheduled, the sequence scheduling information is input under a schedule tab 605 as shown in exemplary Figures 6A-6D. When a sequence is scheduled, the time at which it will be automatically executed is determined by the information supplied on the schedule tab 605. The user selects the frequency at which the sequence is to be executed in the run occurs box 650. The user has the options of more than daily, daily, weekly, or monthly. Other options may be provided as well. Regardless of which option is chosen, the user enters a begin time 610 and a limit time 630. The begin time 610 indicates the time of day that the sequence is to be run. A begin date 620 to be used in connection with the begin time 610 is also established. A limit time 630 is the maximum amount of time that the sequence can be run before some intervention will be taken by the scheduler. When the sequence exceeds the limit time, a desired action to be taken may be selected from a drop-down menu 635. Some over-limit actions include sending messages to the error log to allow the sequence to continue, to stop the sequence and reschedule, to stop the sequence or remove it from the schedule, or to ignore the over-limit.

The frequency of execution of a sequence may be selected in a run occurs box 640. In one embodiment, the run occurs box 640 includes a more than daily button 641, a daily button

642, a weekly button 643, and a month button 644. If the sequence is selected to run more than once per day by clicking the more than daily button 641, a more than daily box 645 as shown in Figure 6A is displayed to allow input of a delta time 650. The delta time 650 is the amount of time between scheduled executions. The sequence adds the delta time 650 to the begin time 610 to determine a subsequent start time for the sequence. For example, if a begin time of 12:00 and a delta time of 5 minutes is selected, the sequence is set to execute every 5 minutes beginning at 12:00. This results in the sequence executing 12 times an hour at 5 minutes past the hour, 10 minutes past the hour, 15 minutes past the hour, and so on. The action the sequence should take if the system does not start within the delta time 650 may be specified in the over-delta action box 655. These actions are similar to the actions in the over-limit action box 635. Finally, the next scheduled run time 660 of the sequence is displayed in the more than daily box 645. This provides a constant indication of when the sequence is scheduled to run next.

If the daily option is selected in the run occurs box 640 by clicking on the daily button 643, the daily option box 665 appears as shown in Figure 6B. The number of days between executions of the sequence may then be selected. For example, if one day is entered in a frequency box 670, the sequence would execute at the begin time 650 every day starting with the begin date 620. If the value entered in the frequency box 670 is greater than one, the sequence will be executed on the day designated as the begin date 620 and then every N days after that where N is the number of days entered.

If the weekly option is selected in the run occurs box 640 by clicking on the weekly button 643, the weekly selection box 675 as shown in Figure 6C is displayed. How often on a weekly basis the sequence runs may then be selected using a weekly frequency box 680. For

example, if one week is entered in the weekly frequency box 680, the sequence runs on the same days each single week. The days of the week on which the sequence run is to occur is selected using checkboxes 685. To run the sequence on a less frequent scale, the number in the weekly frequency box may be increased. This results in the sequence running only every N weeks on 5 the specified days. The weekly frequency box 675 also includes an indication of the next scheduled run time 660 based on the frequency selected.

If the monthly option is chosen in the run occurs box 640 by clicking the monthly button 644, the monthly frequency box 690 as shown in Figure 6D is displayed. In this situation, there are two choices in how to specify which day of the month to run the sequence. When a day option 691 its selected, a particular day 692 of the month for the sequence to run is entered. An option also exists to select a monthly frequency 694 for the sequence to run. The second option is very flexible, allowing the first, second, third, fourth, or last particular day, i.e., Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday, of some number of months to be selected. The first, second, third, fourth, or last day is selected from a day frequency pick list 15 696. The particular day to run the sequence is selected from a day pick list 697. The monthly frequency window 698 allows a sequence to run every N number of months. As with the other sequences, the next scheduled run time 660 is displayed to the user.

In addition to scheduling the sequences, each task in a sequence may be defined by double clicking on the selected task in the sequence window 340, thereby displaying a task 20 dialog box 700 for the task, one embodiment of which is shown in Figure 7. As described above, each sequence is composed of a list of tasks. Each task is associated with a separate task dialog box 700. In each task dialog box 700, the title bar 702 contains the name of the task. Below the

title bar, general information about the task appears including the task type and task description.

Under a general tab 705, the task can be designated as activate or inactivate by selecting an appropriate activation level, either active or inactive, from a status box 710. When inactive, the task is bypassed in the sequence and the next task following the "continue exit branch" of the

5 inactive task is designated to be executed next.

The task dialog box 700 also contains a message logging box 715. The message logging box 715 establishes the level of message logging to be performed by the task. The task may be set to default to the same level of message logging as is set for the entire sequence by clicking on the default to sequence option, or the task may override the sequence setting with a task-specific level from the pick list 720 by clicking on a level option. Types of message logging levels are described above with respect to the message logging box 515.

The task dialog box 700 also contains an execution limit box 725 to limit the amount of time allowed to execute the task. The maximum amount of time for the task to execute can be entered in the limit time box 730. If a limit time 730 is specified, an over-limit action should 15 also be specified from the pick list 735. The over limit actions include, by way of example, logging an error message and continuing, aborting the task and continuing the next task, or stopping the entire sequence. A notes tab 707 in the task dialog box 700 may be selected to allow the user to enter documentation associated with the particular task. These may include a short description of the task or detailed notes about the task.

To provide for maximum flexibility, custom tasks may be created and included in the sequence. If a custom task is selected from the list of task icons in the task palette 320, the custom task dialog box 800 appears as shown in Figure 8. By selecting a script tab 805, a script

defining the task may be input. In one embodiment of the invention, the Tool Command Language (TCL) is used. Of course, other scripting languages may be used without departing from the spirit of the invention. The task script defines the instructions to be run when the task is executed. Within the task script, any of the published methods defined for any of the objects in 5 the system may be called. A custom task also has access to any custom data defined for the task. At the completion of the task, the custom task may select an appropriate exit branch. In one example, a custom task could be programmed to provide a visual indication to the user when the plant data is optimized.

10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
9185
9190
9195
9200
9205
9210
9215
9220
9225
9230
9235
9240
9245
9250
9255
9260
9265
9270
9275
9280
9285
9290
9295
9300
9305
9310
9315
9320
9325
9330
9335
9340
9345
9350
9355
9360
9365
9370
9375
9380
9385
9390
9395
9400
9405
9410
9415
9420
9425
9430
9435
9440
9445
9450
9455
9460
9465
9470
9475
9480
9485
9490
9495
9500
9505
9510
9515
9520
9525
9530
9535
9540
9545
9550
9555
9560
9565
9570
9575
9580
9585
9590
9595
9600
9605
9610
9615
9620
9625
9630
9635
9640
9645
9650
9655
9660
9665
9670
9675
9680
9685
9690
9695
9700
9705
9710
9715
9720
9725
9730
9735
9740
9745
9750
9755
9760
9765
9770
9775
9780
9785
9790
9795
9800
9805
9810
9815
9820
9825
9830
9835
9840
9845
9850
9855
9860
9865
9870
9875
9880
9885
9890
9895
9900
9905
9910
9915
9920
9925
9930
9935
9940
9945
9950
9955
9960
9965
9970
9975
9980
9985
9990
9995
10000
10005
10010
10015
10020
10025
10030
10035
10040
10045
10050
10055
10060
10065
10070
10075
10080
10085
10090
10095
10100
10105
10110
10115
10120
10125
10130
10135
10140
10145
10150
10155
10160
10165
10170
10175
10180
10185
10190
10195
10200
10205
10210
10215
10220
10225
10230
10235
10240

plant history data to assist in optimizing the plant model. An output task is similar to the input task. However, in the output task, the user chooses export and upload options. The e-mail task sends an e-mail message to the designated address. The user enters the e-mail address in the text of the message, to send a message notifying an operator or other program of a critical failure or 5 other designated message. The load case task is used to load a case or set of data into the flow sheet of the associated model application. The store case task allows the user to store the data currently loaded in the flow sheet. The solve task allows the user to solve a case, e.g., a simulation or optimization problem, that has been defined for the flow sheet.

The calculation mode can be selected or a default calculation mode may be used. Advantageously, a single flowsheet configuration is enabled to be used in multiple calculation modes. In one preferred embodiment, three calculation modes, simulation, data reconciliation, and optimization, are provided. The calculation modes refer to alternative flowsheet solution-types that are designed to accomplish different solution objectives. For example, a single flowsheet model may generate an open-form equation representation for use in simulation, an 15 open-form equation representation for use in data reconciliation, and an open-form equation representation for use in optimization.

The data reconciliation pre-processing task performs the steps necessary to prepare a data reconciliation case for the flow sheet to be solved. The data reconciliation review task reviews the solution of a data reconciliation case and determines what task to perform next based upon 20 the results. The optimization pre-processing task performs the pre-processing steps necessary to set up an optimization case for the flow sheet. The optimization review task reviews the results of an optimization solution. The implementation pre-processing task performs the pre-

processing steps necessary before sending targets to the controllers. The model sequence activation control task controls the activation of on-line model sequences that are not scheduled. The activation is based on various criteria that is set up for each on-line model sequence.

A steady state detection task determines if the unit is steady or unsteady by monitoring

- 5 the values of a set of process measurement points. Selecting a steady state detection task causes a steady state detection task dialog box 1000 to be displayed as shown in Figure 10. The circumstances which define steady state can therefore be varied in each sequence. In the steady state detection task dialog box 1000, the number of periods to be monitored for steady state is set in a period monitoring selection box 1010. For example, a history of the last N (where N is the number of periods entered by the user) measure values of each point is obtained and a statistical test is performed to determine whether the value of the point has significantly changed. If the point has not significantly changed, the value is determined to be steady. The minimum percentage of individual points needed to be steady for the overall unit to be considered in steady state can be specified in a threshold box 1015. In a results box 1020, the average of all points
15 "percent steady" values is displayed in a percent steady box 1025. The percent steady box 1025 value is compared with the minimum percent required for steadiness as entered in the threshold box 1015, and the final result is given to the user as steady or unsteady in a steadiness indication box 1030.

- One example of a sequence 1100 created by one embodiment of the present invention is
20 illustrated in Figure 11. The sequence 1100 begins as indicated in a start state 1105. Proceeding to state 1110, the sequence load a previously configured solution of a plant model. The specifics

of the layout and operation of the plant are included in the plant model. The plant model may be the result of a series of previous sequences.

Proceeding to state 1115, the sequence 1100 inputs new plant data. The data can be input by a user from the keyboard or may be downloaded from historical data. The sequence then 5 proceeds to state 1120, where the software attempts to solve the plant model.

Proceeding to state 1120, the solution from state 1120 is checked to determine if the solution is valid. If the solution is valid, the sequence 1100 proceeds along the YES branch to state 1130. In state 1130, the sequence save the solution and initializes the software for the next run. The sequence 1100 then proceeds to end state 1145.

10 Returning to state 1125, if the solution is not valid, the sequence 1100 proceeds along the NO branch to state 1135 to determine if more iterations are likely to produce a valid solution. If there is an indication additional iterations might produce a valid solution, the sequence proceed along the YES branch back to state 1120. The sequence may remain in this loop until either the solution is valid or a determination is made more iterations are not likely to produce a valid 15 solution.

Returning to state 1135, if it is determined a valid solution is not likely, the sequence 1100 proceeds along the NO branch to state 1140. In state 1140, the case is stored for future analysis. The sequence then proceeds to end state 1145.

Numerous variations and modifications of the invention will become readily apparent to 20 those skilled in the art. Accordingly, the invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The detailed embodiment is to be considered in all respects only as illustrative and not restrictive and the scope of the invention is,

SIMI-003/01US

PATENT

therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

10037786 - 051002